Journal of Intelligent & Robotic Systems
https://doi.org/10.1007/s10846-018-0902-9

Evolutionary Modular Robotics: Survey and Analysis

Reem J. Alattas[1] · Sarosh Patel[1] · Tarek M. Sobh[1]

Received: 2 October 2017 / Accepted: 3 July 2018
© The Author(s) 2018

Abstract

This paper surveys various applications of artificial evolution in the field of modular robots. Evolutionary robotics aims to design autonomous adaptive robots automatically that can evolve to accomplish a specific task while adapting to environmental changes. A number of studies have demonstrated the feasibility of evolutionary algorithms for generating robotic control and morphology. However, a huge challenge faced was how to manufacture these robots. Therefore, modular robots were employed to simplify robotic evolution and their implementation in real hardware. Consequently, more research work has emerged on using evolutionary computation to design modular robots rather than using traditional hand design approaches in order to avoid cognition bias. These techniques have the potential of developing adaptive robots that can achieve tasks not fully understood by human designers. Furthermore, evolutionary algorithms were studied to generate global modular robotic behaviors including; self-assembly, self-reconfiguration, self-repair, and self-reproduction. These characteristics allow modular robots to explore unstructured and hazardous environments. In order to accomplish the aforementioned evolutionary modular robotic promises, this paper reviews current research on evolutionary robotics and modular robots. The motivation behind this work is to identify the most promising methods that can lead to developing autonomous adaptive robotic systems that require the minimum task related knowledge on the designer side.

Keywords Evolutionary robotics · Modular robots · Task-based design · Self-assembly · Self-reconfiguration · Self-repair · Self-reproduction

1 Introduction

Producing autonomous adaptive robots is a huge challenge. In biology, autonomous and adaptive creatures are produced using evolution. However in industry, mainstream robots use machine learning to produce adaptive behavior to simulate biological aspects while neglecting the autonomous side of it. Therefore, evolutionary algorithms are used to optimize robotic autonomy and adaptation producing what is known as evolutionary robots [1].

Evolutionary robotics evolves populations of robots by applying evolutionary computational methods, and then selects the fittest to survive. The evolutionary approach continuously designs and builds different robots with improved capabilities rather than practicing the hand design approach that can be extremely difficult when designing autonomous adaptive robots. Thus far, applying evolutionary algorithms lack the guarantees of finding an optimal solution; however, the benefits of this method outweigh the cost. The benefits of evolutionary algorithms include the power to improve the parameters and the structure of the robotic control and morphology [2, 3]. In order to maximize that power, modularity could play a role as the basic building block in the robotic system to simplify the implementation and search space because of its discrete nature.

An initial review of the literature was presented in [4], but this paper presents a holistic review of the area, detailed bibliography, and well referenced discussion of the state of the art that can be used as a guideline to understand the trends of this field. The previous work categorized modular robotic systems based on the dominant feature of each robot. It focused on the modular robots while overlooking the evolutionary aspect of these systems. Evolutionary robotics was discussed as a potential technique that can be applied to evolve the robots control systems. On the other hand, this paper focuses mainly on the evolutionary

✉ Reem J. Alattas
ralataas@my.bridgeport.edu

[1] Interdisciplinary Robotics, Intelligent Sensing and Control (RISC) Lab, School of Engineering, University of Bridgeport, 221 University Avenue, Bridgeport, CT 06604, USA

Published online: 16 July 2018

modular robotic systems that applied evolutionary algorithms to improve the resulting modular robotic structure and behavior with an emphasis on applying evolutionary computation to enhance the modular robotic task based design. It also reviews more prototypes that used evolutionary algorithms to solve certain robotic challenges, such as motion planning or changing the robot geometric structure. This paper starts with reviewing some research work done on evolving robotic control in a fixed morphology and co-evolving robotic control and morphology. Then, modular robotics are discussed as a method to design and implement evolutionary robots in the physical world, as advanced technology and rapid prototyping techniques have made these modular robots feasible. Moreover, evolutionary computation can empower modular robots by allowing them to self-assemble, self-reconfigure, self-repair, and self-reproduce. Thereafter, numerous modular robotic applications are analyzed along with their capabilities of performing various evolutionary challenges. Finally, the current state of the art and challenges are discussed.

2 Evolutionary Robotics

In nature, evolution produces heritable changes in the phenotypes of organisms over multiple generations for better adaptation to the environment. In robotics, evolution has been proposed as a nature inspired approach to avoid the bias and limitations introduced by human designers and to produce better adapted robots to the environmental changes [5]. Simply, evolutionary robotics is a method of creating autonomous robots automatically without human intervention [6].

The Darwinian theory of evolution inspired evolutionary robotics. This theory states that all organisms develop through mutation, crossover, and selection that increase the new generation's ability to compete, survive, and reproduce [7]. Based on the principle of selective reproduction of the fittest, robots are viewed as autonomous artificial organisms that can develop their own skills by interacting with the environment and without human intervention. The fittest robots survive and reproduce until a robot that satisfies the performance criteria is produced [8].

Each robot comprises two major parts: control (brain) and morphology (body). Controls are represented in many ways including neural networks that map sensory input to actuator outputs. Morphology can be described as tree-based representation, L-system consisting of set of rules that can produce construction sequences or regulatory networks. To allow for open-ended synthesis, both control and morphology should co-evolve along with the fitness functions and evaluation methods [9].

Evolving robotic controls has received a lot of attention on research because controls are more adaptable than morphology. Floreano et al. described evolving a small-wheeled robot control that is comprised of a neural network using a simple Genetic Algorithm to navigate a looping maze. Their experiment showed that the fitness function evolved and the cruising speed of the robot evolved as well, demonstrating that evolution can lead to better adaptation [6]. Nolfi and Floreano presented a set of navigational experiments in their book, ranging from the simple to the very complex in order to address different adaptation mechanisms. In some cases, the evolved solution outperformed the hand-designed solution by capitalizing on interactions between machine and environment that could not be captured by a model-based approach. On the other hand, more complex tasks exposed the limits of reactive architectures [8].

In the previous work, robotic controls were evolved for fixed morphological structures that were user-designed. Other research studies indicate the need to co-evolve the robotic control and morphology in order to produce fitter robots, as is the case of nature. Paul and Bongard introduced coupled evolution of robotic morphology and control on a biped robot in simulation. The closed loop recurrent neural network controller was optimized simultaneously with the morphological parameters using a fixed length Genetic Algorithm [10]. Zykov et al. applied the same theory on a physical robot to evolve the dynamic gates in hardware. The nine-legged robot's open-loop controller was evolved using a Genetic Algorithm to allow evolving speed and locomotion pattern under the rhythmicity constraint [11]. Lund et al. investigated the co-evolution of robotic controller and morphology using LEGO parts to construct the evolved morphology and downloaded the evolved controller to LEGO MINDSTORM RCX. The search space for morphology was limited, but the solution search space was enlarged when co-evolving controller and morphology [12, 13]. Sims created a system where virtual robotic control and morphology co-evolve to allow the robots to compete in a physically simulated 3D world to gain control over common resources. The robots were made of 3D cubes and oscillators [14].

The techniques presented for coupled evolution of robotic control and morphology allowed the automation of designing complex systems that would be difficult to design using traditional methods. Lipson and Pollack explored automatic design concepts by building robots using lower-level building blocks with no sensors. The control was composed of neurons and the morphology was composed of bars and linear actuators. The resulting solutions were remarkably elaborate and would have been difficult to design using traditional methods [15]. However, an obvious constraint was the manufacturability of the resulting solutions. Therefore, Faíña et al. proposed the use of modular robots as the fundamental building blocks for evolutionary processes because modularity allows building a wide variety of robotic structures, simplifies the search space, and ensures easy implementation in reality [5].

3 Modular Robotics

Modular robots are composed of various units or modules, hence the name. Each module involves actuators, sensors, computational, and communicational capabilities. Usually, these systems are homogeneous where all the modules are identical; however, there could be heterogeneous systems that contain different modules to maximize versatility [5].

Modular robotic systems have three promises: versatility, robustness, and low cost. Versatility is the capability of the modular robotic system to form a number of different shapes, each with big numbers of degrees of freedom (DOF). In other words, to allow the robot to self-reconfigure in order to accomplish various tasks in different environments. Versatility can be measured by the number of isomorphic configurations the robotic system can form and by the number of DOF in the system. The number of configurations grows exponentially with the number of modules and the number of DOF grows linearly with the number of modules. Robustness comes from redundancy and self-repair. When the robot is composed of many identical modules and one fails, any other module can replace it to keep the system running. Finally, low cost promise is achieved through batch fabrication. As the number of repeated modules increases, the economies of scale come into play and the per-module cost goes down [16]. Also, maintaining low cost can be achieved through rapid prototyping equipment techniques; such as 3D printing, that can build any object by laying down successive layers of material. In order to empower the aforementioned characteristics in modular robotic systems, evolutionary task-based design approaches are utilized to replace traditional design methods.

In 1995, Chen and Burdick proposed a system to find optimal modular robotic design to accomplish a specific task. This system is formulated as a discrete optimization function based on assembly incidence matrix to represent the robotic configuration. These matrixes are encoded into bit strings to best utilize Genetic Algorithm. Genetic Algorithm is used for optimization because of the discrete nature of the search space, even though the application of Genetic Algorithm could be computationally expensive. The robots consist of link and joint modules and were implemented in simulation [17].

Chung et al. introduced in 1997 a task based design method for modular robot manipulators. The robotic system consists of manipulator base, link, and joint modules. The robot configuration is determined using kinematic relations. Then, Genetic Algorithm is used to find the optimal link length for a specific task. Because of the complexity of the problem, the computation effort required for Genetic Algorithm is tremendous. This algorithm has been implemented on physical robots. This work considers only the first level of modular architecture; which is kinematics synthesis [18].

In the same year, Chocron and Bidaud presented an adaptive multi-chromosome evolutionary algorithm for optimizing task based kinematic design of modular robotic systems. The robot consists of a mobile base and a set of link and joint modules to assemble the manipulator arm. The task is specified as 3D end-effector configurations. The robots were implemented in simulation. This method yielded better results compared to two-level GA and multi-chromosome evolutionary algorithm; however, it still lacks optimality. Additionally, the increase in the number of design variables increases the search space exponentially that results in making this algorithm insufficient [19].

According to Yang and Chen, with fewer DOF, the modular robot can better perform the task in terms of energy consumption and loading capacity. Therefore, they proposed the minimized degree-of freedom concept for task based modular robot design optimization in 2000. This system uses assembly incidence matrix to represent the robot configuration as in [17]. It also uses Evolutionary Algorithm (EA) to search for optimal solutions. This algorithm was implemented in simulation and produced sub-optimal results [20].

Hornby et al. proposed in 2001 an automatic design system for modular physical robots that can become more complex. The goal of their work is to overcome the limitations of using evolutionary design approach to make it ready for practical engineering by reaching high complexities and to simplify design changes by using generative design and allowing reusability of modules. The robots consist of bars and joints inspired by Tinker-Toy™ components. The 2D robots are produced in simulation using Lindenmayer system for design and evolutionary algorithm for optimization. Then the actual robot is hand-assembled from an evolved design. This work described the co-evolution of the robotic control and bodies for locomotion abilities and produced a real robot that was tested and moved in the real world [21].

In 2013, Faíña et al. proposed an evolutionary designer for heterogeneous modular robots. This system is capable of designing complete robots including their control and morphology. Each robot involves distributed control and heterogeneous modular architecture. Tree like representation is used for robotic morphology to smooth the search space. A constructive evolutionary strategy is used to co-evolve the robots control and morphology in simulation. Then the resulting robots are implemented physically to prove feasibility. The results were promising when applied to solve a linear motion problem but needed improvements in the case of static problems [5].

Moreover, evolutionary algorithms can be applied to modular robots to allow self-assembly from constituent modules, self-reconfiguration into different functional forms, self-repair to detect errors and recover from failures, and self-reproduce where one system can produce another autonomous functional system.

3.1 Self-Assembly

One of the main benefits of modularity is the capability of self-assembly, which is the natural construction of complex multi-unit system using simple units governed by a set of rules. The self-assembly process is ubiquitous in nature as it generates much of the living cell functionality [22]. However, it is uncommon in the technical field because it is considered as a new concept relatively in that arena, although it could help in lowering costs and improving versatility and robustness; which are the three promises of modular robotics. The ability to form a larger stronger robot using smaller modules allows self-assembled robots to perform tasks in remote and hazardous environments.

In other words, self-assembly is the problem of designing a collection of elements with edge binding properties such that, when they mix randomly, they bind to form desired assemblies. The elements may be homogenous or heterogeneous; their binding properties may be fixed or dynamic; and they may have a range of capabilities such as ability to detect binding events or exchange information with neighbors [23].

Various types of evolutionary algorithms have been utilized in implementing self-assembly modular robotic systems. Bonabeau et al. studied employing Genetic Algorithm to generate self-assembly rules for modular robotic systems and explored the relationship between the space of possible rules and resulting biologically plausible structured architectures. This work did not address the problem of self-assembly according to pre-determined shapes [24].

Tolley et al. extended the stochastic self-assembly modular robot proposed in [25] from 2D to 3D. They used evolutionary approach to design robotic structures according to an input function. These structures are evolved in simulation using frequency-based representation. Then the assembly algorithm takes place to plan the assembly of the fittest evolved robot by sampling a graph of all possible paths to the target structure and following those that leave the most options open. The modules in this system are unable to move on their own because they need to circulate in turbulent fluid to accrete onto the structure. This fluidic system could be scaled down to produce micro-scale modules. This system lacks the possible feedback between the design and assembly phases, which can have a large impact on the evolving design if implemented to adapt to assembly conditions [26].

3.2 Self-Reconfiguration

Recently, modular robotics has gotten attention from researchers in the robotics field because of their ability to self-reconfigure [27]. Modular self-reconfigurable robots involve various modules that can combine themselves autonomously into meta-modules that are capable of performing various tasks under different circumstances [5]. The ability to self-reconfigure allows these robots of metamorphosis, which in turn makes them capable of performing different sorts of kinematics. For instance, a robot may reconfigure into a manipulator, a crawler, or a legged one [27]. This sort of adaptability enables self-reconfigurable robots to accomplish tasks in unstructured environments; such as space exploration, deep sea applications, rescue missions, or reconnaissance [28].

Yim et al. in 2002 classified reconfigurable robots into three classes of architecture: lattice, chain, and mobile based on how they reconfigure [28]. Then they added deterministic and stochastic reconfigurations in 2007 [29].

Lattice architectures have modules that are arranged in a 2D or 3D grid that can be used as a guide for modules to determine their positions and form the new shape accordingly. All modules remain attached to the main body to simplify planning and control [29]. Moreover, lattice architectures are capable of offering simpler reconfiguration compared to other classes because control and motion can be executed in parallel [27]. Lattice-type systems exploit lattice regularity when aligning connectors during self-reconfiguration in order to allow faster and easier self-reconfiguration. However, assuming that all modules conform to the lattice can be problematic for systems with a big number of modules [30]. One example of a lattice-based self-reconfigurable robot is M-TRAN [31, 32].

Chain/Tree architectures have modules that are connected together in a string or tree topology. The serial underlying architecture implies that each chain is always attached to the rest of the modules at one or more points, and the modules reconfigure by attaching and detaching to and from themselves. The chains may be used as robotic arms, legs, or tentacles [29]. Chain architectures are more versatile compared to other architectures due to their capability of reaching any point in space through articulation, but they are more difficult to control and more expensive computationally to represent and analyze [27]. An example of a chain-based self-reconfigurable robot is PolyBot [33–35].

It is important to mention that lattice architecture and chain architecture do not contradict, and numerous systems can be of both types at the same time, such as SuperBot [36] and UBot [37]. These systems tend to have Hybrid architectures [28].

Mobile architectures have modules detach from the main body and maneuver independently using the environment; e.g. liquid or outer space, to link up at new locations in order to form new shapes, complex chains or lattices, or form a number of smaller robots. Mobile architecture is less explored compared to other structures because the reconfiguration difficulty outweighs the functionality gain [28, 29]. A mobile-based self-reconfigurable example system is CEBOT [38–40].

Deterministic Architectures have modules move directly to their target locations during the self-reconfiguration process. Each unit's location can be known at all times or calculated at run time, such that reconfiguration times are guaranteed. Feedback control is necessary to ensure precise movement. Usually, macro-scale systems are considered deterministic [29].

Stochastic Architectures have modules move in a 2D or 3D environment using statistical processes; e.g. Brownian motion, which are used to guarantee reconfiguration times. The exact location of each unit is known only when it is connected to the main structure, but the paths taken by those units to move between locations might be unknown. Stochastic architectures are more ideal at micro-scale systems. The environment provides most of the needed energy for moving units around [29].

Evolutionary algorithms were used to evolve modular self-reconfigurable robotic controls in order to support self-reconfiguration and also to implement different modular robotic behaviors. Østergaard and Lund explored evolving controllers of M-TRAN [41] and ATRON [30] self-reconfigurable modular robotic systems in simulation. Employing Genetic Algorithm for implementing M-TRAN walking behavior is very complicated because evolving each controller locally to generate a global behavior is affected by the conditions of neighboring modules. Therefore, when attempting to evolve ATRON controllers individually to allow the modular collection of moving in the right direction, two modules were evaluated as a couple instead of evaluating single modules using competitive co-evolution and symbiotic co-evolution. This work did not address the constraints of physical systems [30].

ACMoD is a modular self-reconfigurable robot that uses Genetic Algorithm to produce proper configuration patterns and for optimizing the path of modules through a static grid of different terrain blocks. This work did not address dynamic environment or found optimal solutions. The system was implemented in simulation [42].

3.3 Self-Repair

Self-repair is a special type of self-reconfiguration that allows a robot to replace damaged modules with functional ones in order to continue with the task at hand [27]. A self-repair system must have two qualities: the ability to self-modify, and the availability of new parts or resources to fix broken ones. Therefore, modular self-repair robots usually consist of redundant modules. Self-repair involves two phases: detecting the failure module, and then ejecting the deficient module and replacing it with an efficient extra module. Such robots are well suited for working in unknown and remote environments.

ATRON modular robot uses evolutionary algorithms to implement self-repair functionality [30]. This system is discussed in detail in the Applications section in this article.

3.4 Self-Reproduction

The ultimate form of self-repair is self-reproduction; which allows robots to reproduce themselves from an infinite supply of parts using simple rules. If the resulting system is an exact replica of the original, the system is called a self-replicator [43]. The effort in self-reproducing is focused on the design and construction of a small seed system that grows exponentially to form a larger system through tens of generations. The resulting self-reproducible robots are capable of accomplishing very large-scale tasks, such as collection of solar energy, direct removal of greenhouse gases from the Earth's atmosphere, and water purification for irrigation. Self-reproduction differs from self-assembly because the resulting systems do not need to make copies of themselves in the latter cases. Since any replication process requires an external material supply, some lattice positions may act as dispensers, where new modules reappear when removed from that location. Self-replication is classified to the following types [41].

- Direct reproduction: A robot picks modules from a dispenser and places them in a new location to gradually build a copy from the ground up.
- Multi-parent reproduction: Multiple robots produce a single copy such that one machine places modules, while the other assembles these modules.
- Self-assisted reproduction: The robot being built self-reconfigures to assist its own building during the building process.

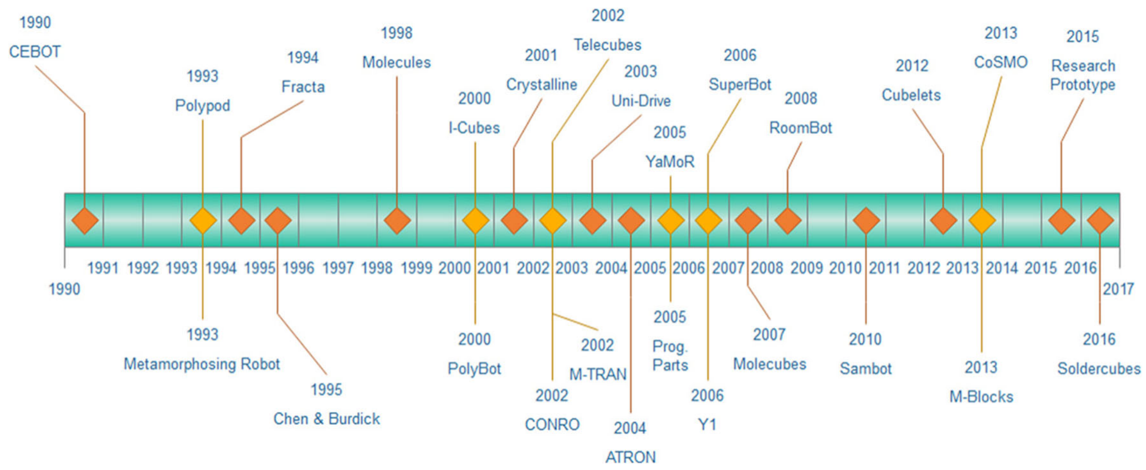


Fig. 1 Chronogram of selected modular robotic prototypes

- Multi-stage reproduction: Temporary scaffold is needed in order to build the target robot. Then this temporary scaffold is either discarded as waste or re-used to produce additional robots.

Self-replicators can use evolutionary algorithms to evolve into the goal structure. The evolution occurs in two stages: morphology evolution and control evolution. Zykov et al. used Genetic Algorithm to evolve Molecubes in 2D simulation using two distinct fitness functions: one for evaluating the fitness of morphology and the other to evaluate the fitness of control. The robotic structure was expressed using a variable-length genome and the control was described using command sequence. Only few results were successful yielding separate identical copy and a matching control. The successful results were implemented on physical robots. This work faces a computational challenge in the planning of self-replication algorithms [43].

Table 1 Modular robotic systems classification based on holistic system characteristics

	Self-assembly	Self-reconfiguration	Self-repair	Self-replicate
PolyBot		√		
I-Cubes		√		
Crystalline		√	√	
Telecubes		√		
CONRO		√		
M-TRAN		√		
Uni-Drive				
ATRON		√	√	
Programmable Parts	√			
YaMoR		√		
Y1				
SuperBot		√		
Molecubes		√		√
RoomBot	√	√		
Sambot	√	√		
Cubelets				
M-Blocks	√	√		
CoSMO		√		

4 Applications

There is a growing number of modular robotic prototypes that has been studied in the literature, so this section reviews

Table 2 Modular robotic systems classification based on modularity state of matter

	Homogeneous	Heterogeneous
PolyBot	√	
I-Cubes		√
Crystalline	√	
Telecubes	√	
CONRO	√	
M-TRAN	√	
Uni-Drive		
ATRON	√	
Programmable Parts	√	
YaMor	√	
Y1	√	
SuperBot	√	
Molecubes	√	
RoomBot	√	
Sambot	√	
Cubelets	√	
M-Blocks	√	
CoSMO	√	

Table 3 Modular robotic systems classification based on implementation method

	Simulation	Physical implementation
CEBOT	√	
Polypod		√
Metamorphosing Robot	√	
Fracta		√
Chen & Burdick Robot	√	
Molecules		√
PolyBot		√
I-Cubes		√
Crystalline		√
Telecubes		√
CONRO		√
M-TRAN		√
Uni-Drive		√
ATRON		√
Programmable Parts		√
YaMor		√
Y1		√
SuperBot		√
Molecubes		√
RoomBot		√
Sambot		√
Cubelets		√
M-Blocks		√
CoSMo		√

a number of emphasized prototypes that participated in the growth of evolutionary modular robotics research.

The timeline covered in this paper ranges from 1990 until this year. Figure 1 illustrates a chronogram of some of the surveyed systems along with other systems that were not covered in detail in this paper. Tables 1, 2 and 3 compare some of those systems based on different parameters.

Figure 2 demonstrates some of the physical systems. The subsections below are chronologically ordered by publication date.

4.1 PolyBot – 2000

PolyBot is a modular self-reconfigurable robot that was implemented to explore how realistic it is to make robots using several homogeneous hardware modules. Three generations of PolyBot modules were prototyped, such that each generation addresses a number of shortcomings discovered in the previous one. The first generation (G1) is constructed using two modular types: node and segment. The segment modules are nominally rectangular prisms with 1 rotational DOF separating two connection ports. The node modules are fixed passive cubes with 6 connection ports. Unlike its G1 predecessor, the second generation (G2) connection ports have electromechanical latches under software control. These latch onto the pins protruding from the opposite face. The third generation (G3) modules are smaller and lack the DC motor extending past the side of each module. The new module has instead a DC pancake motor with a harmonic gear that is completely internal. The connectors are larger and have higher contact force for higher current loads to enhance performance.

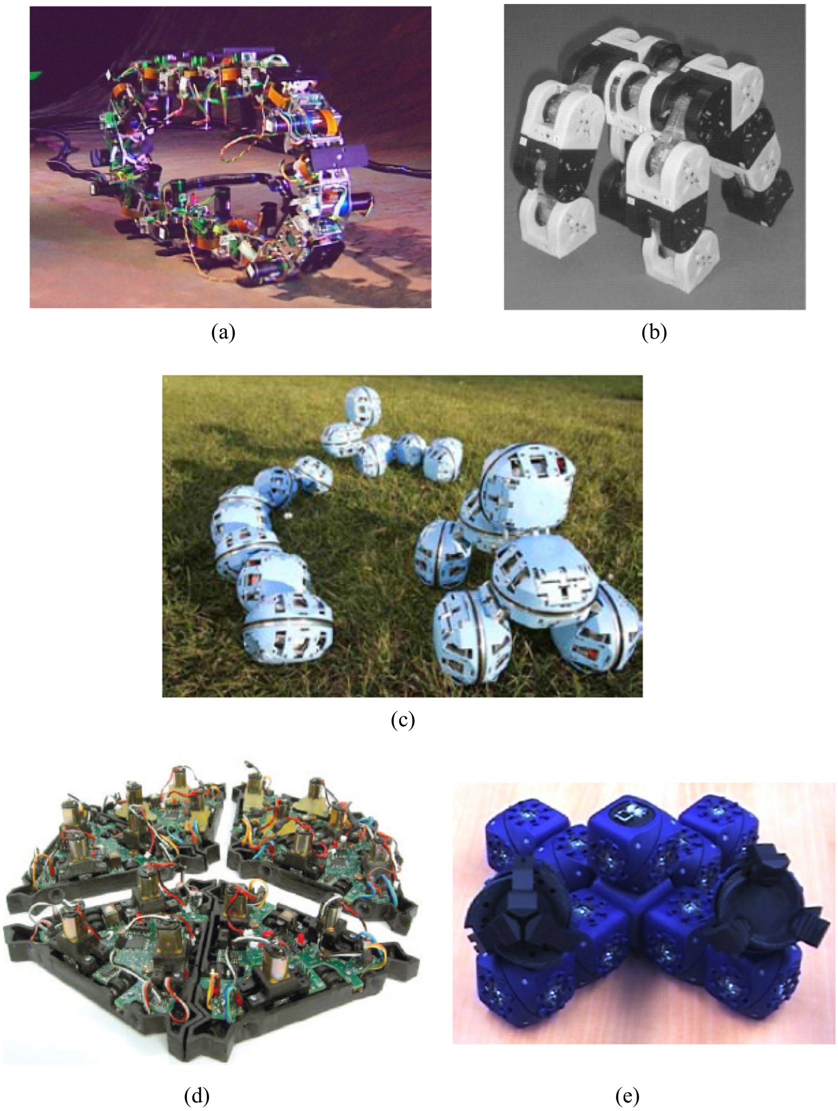
The first two generations of PolyBot prove versatility by executing locomotion over different terrains. However, as the number of modules increases, cost increases, and robustness decreases because of software scalability and hardware dependency problems. Currently the maximum number of modules utilized in one connected PolyBot system is 32 with each module having 1 DOF. The third generation deals with 200 modules to show a variety of capabilities, including moving like a snake, lizard, or centipede as well as humanoid walking and rolling in a loop [33–36].

PolyBot is capable of self-reconfiguration by changing its geometry and locomotion mode depending on the terrain type – rolling over flat terrain, earthworm to move around obstacles, and a spider to step over hilly terrain. Planning the self-collision-free motions can be challenging because the size of this space is exponential in the number of modules, but proportional to the number of DOF. For many applications, a fixed set of configurations is sufficient. In this case, reconfigurations can be pre-planned off-line and stored in a table for ease of reconfiguration [16].

4.2 Telecubes – 2002

Telecubes are compact cubic modules that were introduced by Suh et al. as an extension to the Crystalline system [44]. Each cube has 6 prismatic DOF and 4 sides capable of expanding more than twice its original length. These cubes can form a modular self-reconfigurable robot by attaching and detaching magnetically to other cubes [45].

When it comes to reconfiguration, it is assumed the initial and final configurations overlap by at least one meta-module. A module is selected based on the Minimum Manhattan Distance to begin moving. Then, a route is planned for that selected module using a technique similar to the PacMan Algorithm. Once the path is generated, it can be converted into a sequence of motion commands that can be executed. During execution, the meta-modules are divided into active and passive groups. The active modules initiate the planning sequence. The passive modules follow the orders given by active modules to move. This reconfiguration algorithm lacked local decision making and parallel execution [46].



(a)

(b)

(c)

(d)

(e)

Fig. 2 **a** PolyBot G2 [34] **b** M-TRAN III [47] **c** ATRON [30] **d** Programmable Parts [49] **e** Molecubes [53]

4.3 M-TRAN – 2002

M-TRAN (Modular Transformer) is a distributed lattice-based self-reconfigurable robotic system that can metamorphose into various configurations, such as a legged machine generating walking motion. The actual system was built using 10 modules and successfully demonstrated the basic operations of self-reconfiguration and motion generation. In order to drive M-TRAN hardware, a series of software programs has been developed including a kinematics simulator, a user interface for designing configurations and motion sequences, and an automatic motion planner [31].

M-TRAN II is the second prototype where many improvements took place to allow versatile whole body motions and complicated reconfigurations. Those improvements contain a reliable attachment/detachment mechanism, high-speed inter-module communication, on-board multi-computers, accurate motor control, and low energy consumption. The software has been improved as well to verify motions in dynamics simulation and to design self-reconfiguration processes [32].

The third prototype, M-TRAN III, has been developed with an improved connection mechanism. Various control modes including single-master, globally synchronous control and parallel asynchronous control are made possible by using distributed control. Self-reconfiguration experiments using up to 24 units were performed by centralized and decentralized control. System scalability and homogeneity were maintained in all experiments [47].

4.4 ATRON – 2004

Another modular self-reconfigurable robot is ATRON, a lattice-based system consisting of approximately spherical modules, where each sphere is constructed as two hemispheres joined by an infinite revolute joint. Actuation is realized as rotation around an axis diagonally through the sphere, where each module can rotate 360° around the equator. This design allows for a very stable construction around the actuated joint since a relatively large area is available for mechanics. However, the spherical basic module design makes it hard to have large flat surfaces connecting to each other. With spherical modules, connectors need to establish essential point-to-point contacts between modules, which are not desirable because of the high collision probability. The limited mobility of ATRON along with other motion restrictions leads to the use of ATRON meta-module to reduce motion constraints. The meta-module is composed of 3 modules: a body in the center that is connected to two legs.

Modular ATRON control comprises three Artificial Neural Networks; one to decide when to emerge, the second to decide when to stop, and the third to calculate the fitness value of every state in the self-reconfiguration and self-repair processes. Genetic Algorithm is used to optimize the weights of the ANNs. Even though ATRON modules are minimalistic because they have only one actuated DOF, a group of modules was capable of self-reconfiguring in 3D simulation. Similarly, ATRON modules demonstrated self-repair successfully in simulation [30, 48].

4.5 Programmable Parts– 2005

In 2005, Bishop et al. built triangular programmable parts that can be assorted on an air table by overhead oscillating fans to self-assemble into various shapes according to the mathematics of graph grammars. The modules can communicate and selectively bond using mechanically driven magnets, without global knowledge of the full shape. Despite planning to build approximately 100 parts, only 6 parts were built for design simplicity reasons. Those six parts were used in an experiment that showed how these parts react similarly to chemical systems [49]. In addition, Napp et al. provided kinetic rate data measurements to the previous work of graph grammar in order to yield a Markov Process Model [50].

4.6 Y1 – 2006

Gonzalez-Gomez et al. developed three minimal configurations using only two and three Y1 modules. Each of these modules is 72 × 52 × 52 mm having 1 DOF and they are capable of attaching and detaching. Y1 module design is inspired by Polybot G1 modules. Then, eight Y1 modules were used to build a modular worm-like robot that is capable of moving in a straight line using a wave propagation gait. The optimal parameters of these designs that include amplitude and phase are calculated using Genetic Algorithm; given velocity, stability, and power consumption restrictions [51].

4.7 Molecubes – 2007

Molecubes system is an open hardware and software platform for modular robotics that was developed to remove entry barriers to the field and to accelerate progress. The system is composed of modules with 1 rotational DOF. Different types of active modules, such as gripper, actuated joint, controller, camera, and wheel along with a number of passive modules were presented. Each module is a cube shaped with round corners that comprises approximately two triangular pyramidal halves connected with their bases so that their main axes are coincident. Each of the 6 faces of the module is equipped with an electromechanical connector that can be used to join two modules together. Symmetric connector design allows four possible relative orientations of two connected module interfaces, each resulting in different robot kinematics [52].

Genetic Algorithm is used to evolve the modular neural network control of the robots in simulation to generate a certain behavior or motion [53]. In order to achieve self-replication, path planning is done with a gene pool that has been built using evolutionary algorithm [54].

4.8 iMobot – 2010

Ryland and Cheng designed an intelligent self-reconfigurable modular robot with each module having 4 DOF and 6 connection faces. In this robotic system, the individual modules have full mobility unlike all the other systems discussed in this paper where the modules must be connected in a cluster to perform all types of locomotion. For example, M-TRAN module can crawl but it needs a second module to turn. In addition, the iMobot robotic system can perform unique locomotion modes such as driving and lifting into a camera platform [55]. iMobot uses a distributed agent based Genetic Algorithm to search for the optimal genotype that can generate a certain robotic gait [56].

4.9 UBot – 2011

UBot is a modular self-reconfigurable robotic system that is capable of multimode locomotion. The locomotion modes include cross, loop, quadruped and other gaits. UBot is a hybrid system combining the advantages of lattice and chain self-reconfigurable robots. Each module is cube shaped based on one universal joint and has two rotational joints in order to achieve 2 DOF [37, 57].

In 2013, a 3D dynamic simulator was developed to simulate rigid body dynamics and evolve robotic locomotion. Evolutionary Robotics was used to find a gait planner for different robotic structures [58].

4.10 SMORES – 2012

Self-assembling MOdular Robot for Extreme Shape-shifting (SMORES) was designed to become a universal modular robot that improves the versatility of self-reconfigurable robots by allowing the robots to reconfigure in three reconfiguration classes; lattice, chain, and mobile. Each module has four active rotational DOF and two wheels to allow mobile movement [59].

A design framework was developed in 2018 to facilitate configuration design of SMORES robots. The proposed system verifies robotic design validity by detecting conflicting commands and loss of stability. Moreover, this system allows existing robotic structures reusability to create complex robotic designs and behaviors [60].

5 Current State of the Art

More recently, new efforts have been pursued in the field of evolutionary modular robotics. Many tasks have been shown to be achievable, especially with the high number of physically implemented robotic systems. However, the majority of modular robotic behaviors; such as self-reconfiguration and self-repair were implemented in simulation despite the existence of a physical prototype; which can be considered as a reality gap. The reason behind this is the high cost of performing evolution inside physical hardware because of power, communication, and other limitations [61]. Researchers have paid attention to bridge the reality gap using rapid prototyping techniques, as this fabrication method becoming more accessible [4].

Lipson and Pollack tried to bridge the reality gap by proposing an approach based on the use of only elementary building blocks and elementary operators in the design and fabrication process. Elementary building blocks are used to minimize inductive bias and maximize architectural flexibility. Also, these blocks allow the fabrication process to be more systematic and versatile. The pre-assembled machine was fabricated as a whole single unit, with plastic supports to connect the moving parts. These supports broke at first motion. Then, standard stepper motors were snapped in, and the evolved neural network was executed on a microcontroller to activate the motors. Three physical machines; shown in Fig. 3, successfully reproduce the behavior of their virtual ancestors in reality [15].

Auerbach et al. tried to bridge the reality gap by proposing RoboGen, an open-source software and hardware platform that 3D prints evolved modular robots. The software component of RoboGen contains an evolutionary

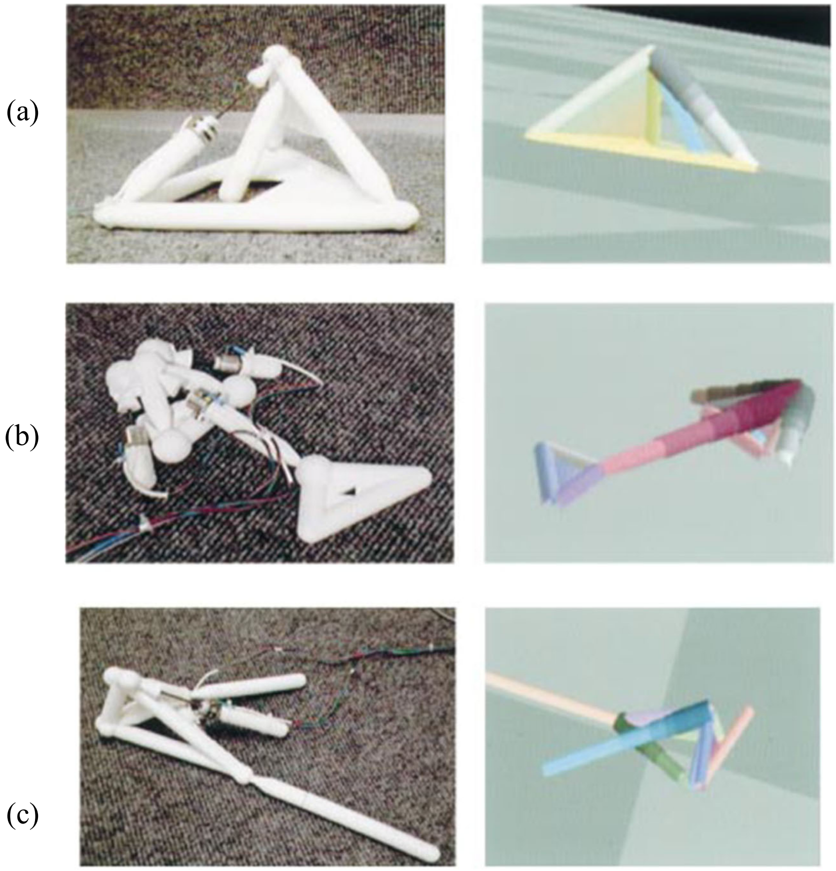


Fig. 3 The resulting robots. Real robots (left); simulated robots (right). **a** Tetrahedron **b** Arrow **c** Pusher [15]

engine to produce modular robots and a simulator to evaluate the fitness of the evolved robots. The robot morphologies are represented as genetic programming trees and the controls are represented as artificial neural networks. Then, the evolved robots can be manufactured using desktop 3Dprinters; as illustrated in Fig. 4 [62].

Additive manufacturing was used by Samuelsen and Glette to build an autonomous robotic system that automatically designs and generates modular robots in simulation using evolutionary algorithms. Then, the system uses an automatic cluster to select the robots with highest fitness values to be manufactured using off-the-shelf motors and 3D printed structural components. The same evaluation procedure was used to compare the performance of the simulated robots and the performance of the physical robots. Five robots were manufactured and three out of five robots have significantly lower performance in reality [63].

Finally, Cellucci et al. demonstrated a 1D printing system inspired by the ribosome to automatically design and fabricate various robots using the same source material. Although the resulting robots have modest functionality, this research points towards expandable robotics that rapidly fabricate customized robots on demand and allow robots recycling to produce new robotic designs that can perform new tasks [64].

The research advances in the field of evolutionary modular robotics could be of interest for the evolutionary mobile robotics community due to the similarities of the challenges facing both fields. Evolutionary mobile robotics applies Evolutionary algorithms to autonomously generate complex robotic behaviors; such as navigation, plant inspection, and rescue operations. Mobile robotic navigation involves path planning as a strategy to find the optimal path from an initial point towards a target point in static and dynamic environments [65]. Evolutionary algorithms have outperformed the classic approaches used to solve the path planning problem because of the NP-hardness of that problem [66]. Among the various meta-heuristic methods belonging to the Evolutionary Algorithms class, Genetic Algorithm has been heavily applied in path planning research.

Qu et al. presented a Co-evolutionary Improved Genetic Algorithm (CIGA) for global path planning of multiple mobile robots that finds a collision-free path for each robot while avoiding collisions between them. The Improved Genetic Algorithm (IGA) is applied to solve the global path planning problem for a single mobile robot, then the co-evolution based on the IGA is used to solve the path planning problem for multiple mobile robots. The IGA involves a modified fitness function, customized selection operator and a new modification operator. The fitness function considers three variables for better accuracy: path length, path feasibility, and path infeasibility. Roulette wheel selection and elitist selection are employed consecutively to ensure high survival probability of the fittest solutions. The modification genetic operator modifies the paths to avoid collision with obstacles and to accelerate the evolutionary process. The system was implemented in simulation and tested in 2D static environment with three mobile robots and a number of static obstacles. The results were near-optimal collision-free paths. This method needs to be improved to include a larger number of mobile robots and involve path planning in dynamic environments [67].

Contreras-Cruz et al. introduced an evolutionary approach to solve the mobile robotic path planning problem. The proposed methodology combines two evolutionary techniques, artificial bee colony and evolutionary programming, to solve the problem in two sequential steps. First, the artificial bee colony algorithm is applied as a local search procedure in order to generate a feasible path. Second, the resulting path is refined by applying the evolutionary programming algorithm to produce short, smooth, and collision-free paths. The proposed method over-performed the classical probabilistic roadmap method in problems with distributed obstacles. However, this method needs to improve the local explorations process. It also needs to consider the multi-robot path planning problem [68].

Lamini et al. developed a new Genetic Algorithm to solve the mobile robotic path planning problem in 2D environments. The proposed method has an improved crossover operator and a new fitness function. The Improved Same Adjacency (ISA) crossover operator takes into

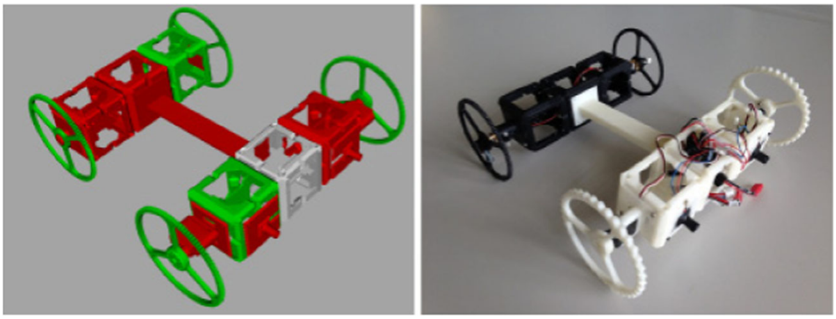


Fig. 4 Evolved robot: simulation (left) and reality (right) [62]

consideration variable length chromosomes to allow rapid convergence and avoid premature convergence. The new fitness function is defined based on the path length, safety, and mobile robotic energy consumption. The system was implemented in simulation and the resulting paths were optimal in terms of the number of turns and iterations [69].

6 Conclusion

This paper has surveyed different applications of evolutionary algorithms in evolving robotic control systems, co-evolving robotic control and morphology, and evolutionary task-based design of modular robots. It was found that evolutionary based solutions exceeded hand designed ones by showcasing novel characteristics and capabilities. The use of evolutionary algorithms in the modular robotic field allowed self-assembly, self-reconfiguration, self-repair, and self- reproduction. These systems were discussed to express how evolutionary robotics can be used to generate rules for planning and controlling general robotic behaviors. Then numerous modular robotic prototypes were analyzed to demonstrate the feasibility of evolutionary modular robotic systems that are capable of accomplishing various tasks in dynamic environments. All of the surveyed systems have physical prototypes; however, the majority of their complex behaviors were implemented in simulation. This leads to a big impediment to advancement in evolutionary modular robotics, which is known as the reality gap. One major solution was reviewed in this paper that has proposed using automatic fabrication to bridge the reality gap. The challenge of building adaptive autonomous robots remains valid, despite the advancements in the field of evolutionary robotics to evolve robotic characteristics and to design modular robots that are capable of performing certain tasks in unstructured environments.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

1. Lipson, H.: Uncontrolled engineering: a review of S. Nolfi and D. Floreano's evolutionary robotics. Artif. Life **4**(7), 419–424 (2000)
2. Bongard, J.C.: Evolutionary robotics. Commun. ACM **56**(8), 74–83 (2013)
3. Silva, F., Duarte, M., Correia, L., Oliveira, S.M., Christensen, A.L.: Open issues in evolutionary robotics. Evol. Comput. **24**(2), 205–236 (2016)
4. Alattas, R.: Analyzing modular robotic systems. In: Online Engineering & Internet of Things, pp. 1014–1028. Springer, Berlin (2018)
5. Faíña, A., Bellas, F., López-Peña, F., Duro, R.J.: EDHMoR: evolutionary designer of heterogeneous modular robots. Eng. Appl. Artif. Intell. **26**(10), 2408–2423 (2013)
6. Floreano, D., Husbands, P., Nolfi, S.: Evolutionary robotics. In: Handbook of Robotics, Chap 61 (2007)
7. Sofge, D.A., Potter, M.A., Bugajska, M.D., Schultz, A.C.: Challenges and opportunities of evolutionary robotics. In: Proceedings of the Second International Conference on Computational Intelligence, Robotics and Autonomous Systems. CIRAS, Singapore (2003)
8. Nolfi, S., Floreano, D.: Evolutionary Robotics: the Biology, Intelligence, and Technology of Self-Organizing Machines. Bradford Books, Cambridge (2004)
9. Lipson, H.: Evolutionary robotics and open-ended design automation. Biomimetics **17**(9), 129–155 (2005)
10. Paul, C., Bongard, J.C.: The road less traveled: morphology in the optimization of biped robot locomotion. In: Proceedings of The IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2001). Hawaii (2001)
11. Zykov, V., Bongard, J.C., Lipson, H.: Evolving dynamic gaits on a physical robot. In: Proceedings of Genetic and Evolutionary Computation Conference GECCO'04 (2004)
12. Lund, H.H., Hallam, J., Lee, W.P.: Evolving robot morphology. In: IEEE International Conference on Evolutionary Computation, pp. 197–202 (1997)
13. Lund, H.H.: Co-evolving control and morphology with lego robots. In: Morpho-functional Machines: the New Species, pp. 59–79 (2003)
14. Sims, K.: Evolving 3D morphology and behavior by competition. Artif. Life **1**(4), 28–39 (1994)
15. Lipson, H., Pollack, J.B.: Automatic design and manufacture of robotic lifeforms. Nature **406**(6799), 974–978 (2000)
16. Yim, M., Duff, D., Roufas, K.: PolyBot: a modular reconfigurable robot. In: IEEE International Conference on Robotics and Automation ICRA'00, pp. 514–520 (2000)
17. Chen, I.M., Burdick, J.W.: Determining task optimal modular robot assembly configurations. In: IEEE International Conference on Robotics and Automation (1995)
18. Chung, W.K., Han, J., Youm, Y., Kim, S.H.: Task based design of modular robot manipulator using efficient genetic algorithm. In: IEEE International Conference on Robotics and Automation, vol. 1, pp. 507–512 (1997)
19. Chocron, O., Bidaud, P.: Evolutionary algorithms in kinematic design of robotic systems. In: Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robots and Systems IROS'97, vol. 2, pp. 1111–1117 (1997)
20. Yang, G., Chen, I.M.: Task-based optimization of modular robot configurations: minimized degree-of-freedom approach. Mech. Mach. Theory **35**(4), 517–540 (2000)
21. Hornby, G.S., Lipson, H., Pollack, J.B.: Evolution of generative design systems for modular physical robots. In: Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation, vol. 4, pp. 4146–4151 (2001)
22. Bishop, J., Burden, S., Klavins, E., Kreisberg, R., Malone, W., Napp, N., Nguyen, T.: Self-organizing programmable parts. In: International Conference on Intelligent Robots and Systems, pp. 3684–3691 (2005)
23. Werfel, J.: Anthills built to order: automating construction with artificial swarms. Ph.D. dissertation, MIT (2006)

24. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: from Natural to Artificial Systems. Oxford University Press, Oxford (1999)
25. White, P.J., Kopanski, K., Lipson, H.: Stochastic self-reconfigurable cellular robotics. In: Proceedings of IEEE International Conference on Robotics and Automation ICRA'04, vol. 3, pp. 2888–2893 (2004)
26. Tolley, M., Hiller, J., Lipson, H.: Evolutionary design and assembly planning for stochastic modular robots. In: Proceedings of the IEEE Conference Intelligent Robotics and Systems (IROS), pp. 73–78 (2009)
27. White, P., Zykov, V., Bongard, J.C., Lipson, H.: Three dimensional stochastic reconfiguration of modular robots. In: Robotics: Science and Systems, pp. 161–168 (2005)
28. Yim, M., Zhang, Y., Duff, D.: Modular robots. IEEE Spectr. **39**(2), 30–34 (2002)
29. Yim, M., Shen, W., Salemi, B., Rus, D., Moll, M., Lipson, H., Klavins, E., Chirikjian, G.: Modular self-reconfigurable robot systems [grand challenges of robotics]. IEEE Robot. Automat. Mag. **14**(1), 43–52 (2007)
30. Østergaard, E.H., Kassow, K., Beck, R., Lund, H.H.: Design of the ATRON lattice-based self-reconfigurable robot. Auton. Robot. **21**(2), 165–183 (2006)
31. Murata, S., Yoshida, E., Kamimura, A., Kurokawa, H., Tomita, K., Kokaji, S.: M-TRAN: self-reconfigurable modular. IEEE/ASME Trans. Mechatron. **7**(4), 431–441 (2002)
32. Kurokawa, H., Kamimura, A., Yoshida, E., Tomita, K., Kokaji, S., Murata, S.: M-TRAN II: metamorphosis from a four-legged walker to a caterpillar. In: Intelligent Robots and Systems (IROS03), vol. 3, pp. 2454–2459 (2003)
33. Yim, M., Eldershaw, C., Zhang, Y., Duff, D.: Self-reconfigurable robot systems: PolyBot.日本ロボット学会誌 21(8):851–854
34. Golovinsky, A., Yim, M., Zhang, Y., Eldershaw, C., Duff, D.: PolyBot and PolyKinetic™ System: a modular robotic platform for education. In: Proceedings of IEEE International Conference Robotics and Automation ICRA'04, vol. 2, pp. 1381–1386 (2004)
35. Duff, D., Yim, M., Roufas, K.: Evolution of polybot: a modular reconfigurable robot. In: Proceedings of the Harmonic Drive International Symposium, Nagano (2001)
36. Yim, M., Zhang, Y., Roufas, K., Duff, D., Eldershaw, C.: Connecting and disconnecting for chain self-reconfiguration with polybot. IEEE/ASME Trans. Mechatron. **7**(4), 442–451 (2002)
37. Zhao, J., Cui, X., Zhu, Y., Tang, S.: A new self-reconfigurable modular robotic system UBot: Multi-mode locomotion and self-reconfiguration. In: 2011 IEEE International Conference on Robotics and automation (ICRA), pp. 1020–1025 (2011)
38. Fukuda, T., Kawauchi, Y.: Cellular robotic system (CEBOT) as one of the realization of self-organizing intelligent universal manipulator. In: IEEE International Conference on Robotics and Automation (ICRA90), pp. 662–667 (1990)
39. Fukuda, T., Nakagawa, S.: Method of autonomous approach, docking and detaching between cells for dynamically reconfigurable robotic system CEBOT. JSME Int. J. Ser. 3, Vib. Control Eng. Eng. Ind. **33**(2), 263–268 (1990)
40. Fukuda, T., Buss, M., Hosokai, H., Kawauchi, Y.: Cell structured robotic system CEBOT: control, planning and communication methods. Robot. Auton. Syst. **7**(2–3), 239–248 (1991)
41. Østergaard, E.H., Lund, H.H.: Evolving control for modular robotic units. In: Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA), pp. 886–892 (2003)
42. Haghzad, S., Bagheri, S., Faraji, S.: Finding proper configurations for modular robots by using genetic algorithm on different terrains. Int. J. Mater. Mech. Manuf. **1**(4), 360–365 (2013)
43. Zykov, V., Mytilinaios, E., Desnoyer, M., Lipson, H.: Evolved and designed self-reproducing modular robotics. IEEE Trans. Robot. **23**(2), 308–319 (2007)
44. Rus, D., Vona, M.: Crystalline robots: self-reconfiguration with compressible units modules. Auton. Robot. (special issue on Modular Reconfigurable Robots) **10**(1), 107–124 (2001)
45. Suh, W., Homans, S.B., Yim, M.: Telecubes: mechanical design of a module for self-reconfigurable robotics. In: Proceedings of IEEE International Conference on Robotics and Automation, vol. 4, pp. 4095–4101 (2002)
46. Vassilvitskii, S., Yim, M., Suh, J.: A complete, local and parallel reconfiguration algorithm for cube style modular robots. In: Proceedings of IEEE International Conference on Robotics and Automation ICRA'02, vol. 1, pp. 117–122 (2002)
47. Kurokawa, H., Tomita, K., Kamimura, A., Kokaji, S., Hasuo, T., Murata, S.: Distributed self-reconfiguration of M-TRAN III modular robotic system. Int. J. Robot. Res. **2**(3–4), 373–386 (2008)
48. Yim, M.: A reconfigurable modular robot with many modes of locomotion. In: Proceedings of the JSME International Conference on Advanced Mechatronics, Tokyo, pp. 283–288 (1993)
49. Bishop, J., Burden, S., Klavins, E., Kreisberg, R., Malone, W., Napp, N., Nguyen, T.: Programmable parts: a demonstration of the grammatical approach to self-organization. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2644–2651 (2005)
50. Napp, N., Burden, S., Klavins, E.: The statistical dynamics of programmed self-assembly. In: Proceedings 2006 IEEE International Conference on Robotics and Automation ICRA, vol. 2006, pp. 1469–1476 (2006)
51. Gonzalez-Gomez, J., Boemo, E.: Motion of minimal configurations of a modular robot: sinusoidal, lateral rolling and lateral shift. Climbing and Walking Robots, pp 667–674 (2006)
52. Zykov, V., Chan, A., Lipson, H.: Molecubes: an open-source modular robotics kit. In: IROS-2007 Self-Reconfigurable Robotics Workshop, pp. 3–6 (2007)
53. Zykov, V., Phelps, W., Lassabe, N., Lipson, H.: Molecubes extended: diversifying capabilities of open-source modular robotics. In: IROS-2008 Self-Reconfigurable Robotics Workshop, pp. 22–26 (2008)
54. Zhong, H., Li, Z., Zhang, H., Yu, C., Li, N.: Modular robot path planning using genetic algorithm based on gene pool. Advances in Computation and Intelligence, pp 380–389 (2010)
55. Ryland, G.G., Cheng, H.H.: Design of iMobot, an intelligent reconfigurable mobile robot with novel locomotion. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 60–65 (2010)
56. Ko, D., Cheng, H.H.: Reconfigurable software for reconfigurable modular robots. In: Proceedings of ICRA 2010 Workshop Modular Robots: State of the Art, p. 100 (2010)
57. Zhao, J., Cui, X., Zhu, Y., Tang, S.: UBot: a new reconfigurable modular robotic system with multimode locomotion ability. Ind. Robot Int. J. **39**(2), 178–190 (2012)
58. Zhu, Y., Zhao, J., Cui, X., Wang, X., Tang, S., Zhang, X., Yin, J. In: IEEE International Conference on Mechatronics and automation (ICMA), pp. 1217–1222 (2013)
59. Davey, J., Kwok, N., Yim, M.: October. Emulating self-reconfigurable robots-design of the SMORES system. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4464–4469 (2012)
60. Tosun, T., Jing, G., Kress-Gazit, H., Yim, M.: Computer-aided compositional design and verification for modular robots. In: Robotics Research, pp. 237–252. Springer, Berlin (2018)
61. Pollack, J.B., Lipson, H., Ficici, S., Funes, P., Hornby, G., Watson, R.A.: Evolutionary techniques in physical robotics. In: International Conference on Evolvable Systems, pp. 175–186 (2000)
62. Auerbach, J., Aydin, D., Maesani, A., Kornatowski, P., Cieslewski, T., Heitz, G., Fernando, P., Loshchilov, I., Daler, L., Floreano, D. Robogen: robot generation through artificial evolution. In: Artificial Life 14: Proceedings of the Fourteenth International

Conference on the Synthesis and Simulation of Living Systems. pp. 136–137. MIT Press (2014)

63. Samuelsen, E., Glette, K.: Real-world reproduction of evolved robot morphologies: automated categorization and evaluation. In: European Conference on the Applications of Evolutionary Computation, pp. 771–782. Springer (2015)
64. Cellucci, D., MacCurdy, R., Lipson, H., Risi, S.: 1D Printing of Recyclable Robots. IEEE Robot. Autom. Lett. **2**(4), 1964–1971 (2017)
65. Nedjah, N., Coelho, L.S., Mourelle, L.M.: Mobile Robots: the Evolutionary Approach. Springer, Berlin (2007)
66. Masehian, E., Sedighizadeh, D.: Classic and heuristic approaches in robot motion planning-a chronological review. World Acad. Sci. Eng. Technol. **29**(1), 101–106 (2007)
67. Qu, H., Xing, K., Alexander, T.: An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots. Neurocomputing **120**, 509–517 (2013)
68. Contreras-Cruz, M.A., Ayala-Ramirez, V., Hernandez-Belmonte, U.H.: Mobile robot path planning using artificial bee colony and evolutionary programming. Appl. Soft. Comput. **30**, 319–328 (2015)
69. Lamini, C., Benhlima, S., Elbekri, A.: Genetic algorithm based approach for autonomous mobile robot path planning. Procedia Comput. Sci. **127**, 180–189 (2018)

Reem J. Alattas Ph.D. candidate in Computer Science and Engineering at the University of Bridgeport, CT. She received the M.S. and B.Sc. with honors in Computer Science from King Saud University in Riyadh, Saudi Arabia in 2007 and 2003, respectively. Reem worked as a Faculty Staff Member (Lecturer) at the Institute of Public Administration and the Arab Open University. Then, she worked for Microsoft Arabia as the first female Technical Account Manager in 2009.

Sarosh Patel assistant professor of Computer Science & Engineering at the School of Engineering, University of Bridgeport. He received his B.Sc. from Osmania Univeristy, India and his M.S. and P.hD. from the University of Bridgeport. His research interests are in the areas of robotics, automation, multi-agent systems and controls.

Tarek M. Sobh received the B.Sc. in Engineering degree with honors in Computer Science and Automatic Control from the Faculty of Engineering, Alexandria University, Egypt in 1988, and M.S. and Ph.D. degrees in Computer and Information Science from the School of Engineering, University of Pennsylvania in 1989 and 1991, respectively. He is currently the Senior Vice President for Graduate Studies and Research, Dean of the School of Engineering and Distinguished Professor of Engineering and Computer Science at the University of Bridgeport, CT; the Founding Director of the Interdisciplinary Robotics, Intelligent Sensing, and Control (RISC) laboratory; the Founder of the High-Tech Business Incubator at UB (CTech IncUBator), and a Professor of Computer Engineering, Computer Science, Electrical Engineering and Mechanical Engineering.